Phoenix Framework

Introduction to Channels

Iuri Fernandes

github.com/iurifq

twitter.com/fqiuri

DIGITAL NATIVES

What is Elixir?

Things people say

- □ “Erlang with a nice syntax”
- □ “The new Ruby”
- □ “Valim’s language”

What is Elixir?

□ Functional language

□ Created by José Valim in 2012

□ Runs on the BEAM (Erlang VM)

□ Influenced by Ruby

José Valim

□ Founder of Plataformatec

□ Rails core commiter alumni

□ Ruby hero 2010

Language Goals

- Extensibility
- Productivity
- Compatibility

Basics

□ Data (types)

□ Modules

□ Processes

Basics

```
defmodule Parallel do
  def map(collection, fun) do
    runner = self
    Enum.map(collection, fn (el) ->
      spawn_link fn ->
        send runner, {self, fun.(el)}
      end
    end)
    |> Enum.map(fn (pid) ->
      receive do
        {^pid, result} -> result
      end
    end)
  end
end
```

Metaprogramming

- Macros
- Access to code representation
  - quote
  - unquote

Metaprogramming

```
defmodule MyMacros do
  defmacro unless(condition, clauses) do
    quote do
      if(unquote(!condition), unquote(clauses))
    end
  end
end
```

Community Response

□ Adopters from Ruby and Erlang

□ Users contributing back

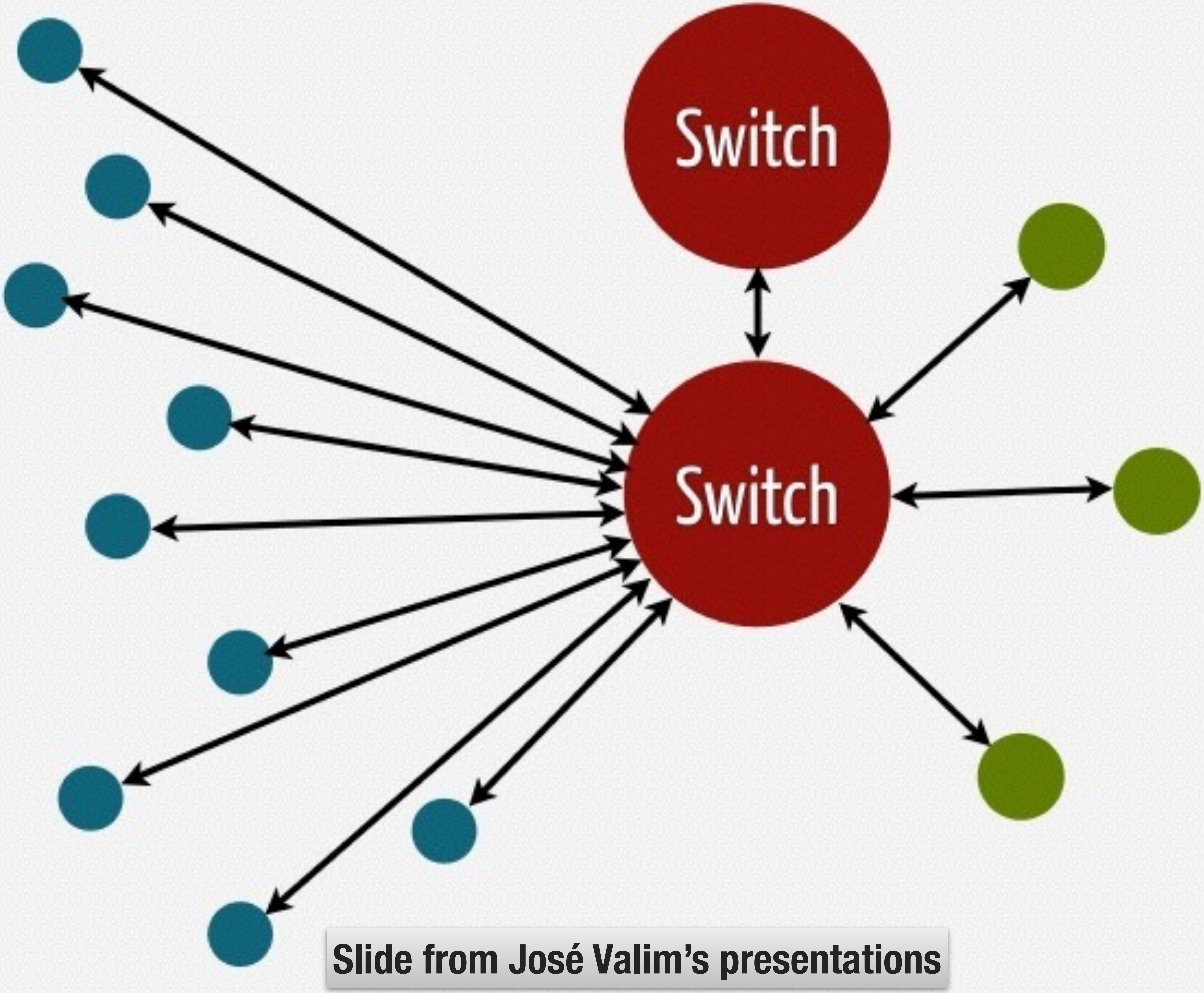
□ Several OSS projects

Awesome Elixir

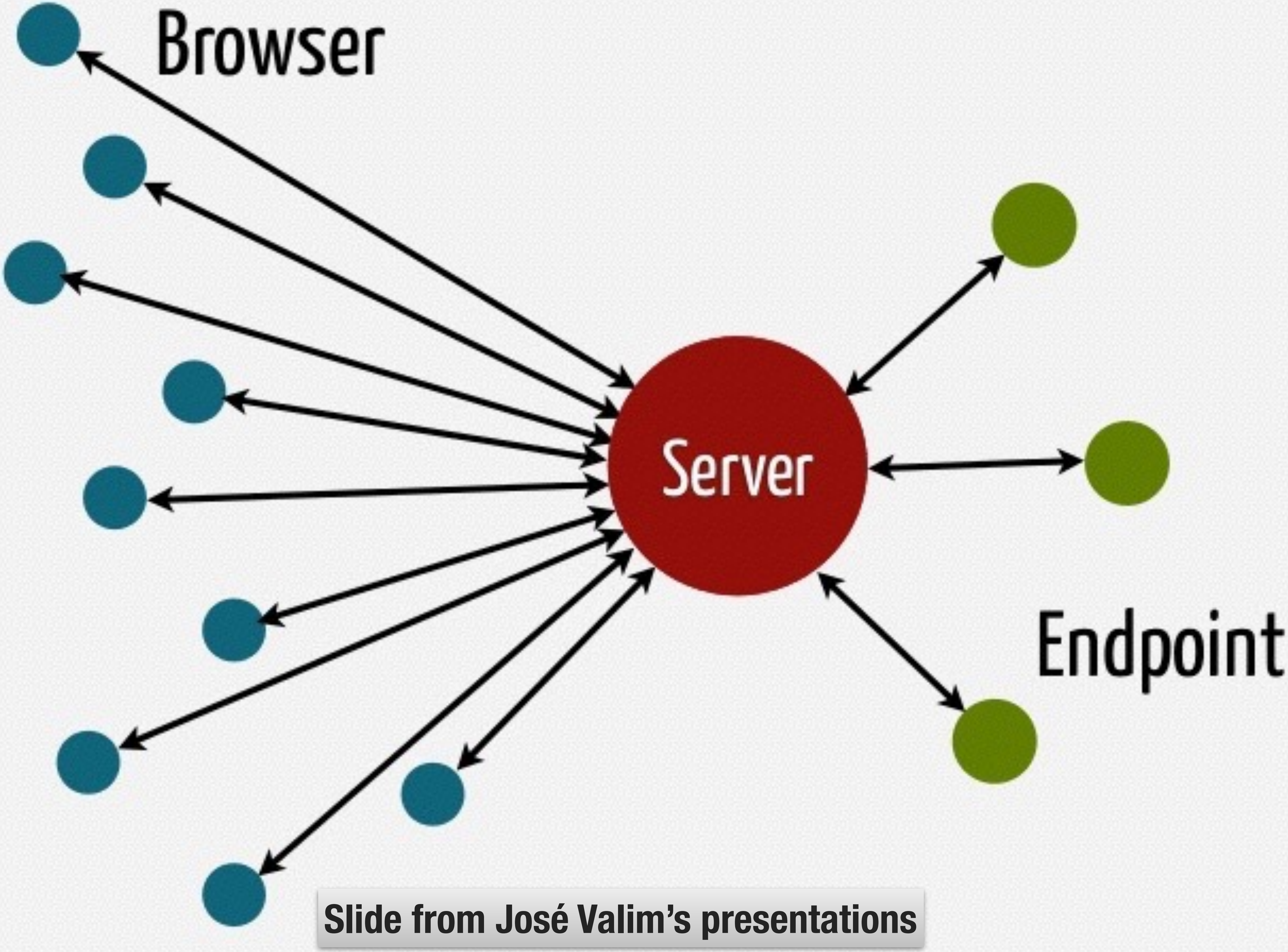
□ Nerves Project

□ Poxa

Why Elixir?



Slide from José Valim’s presentations



Slide from José Valim’s presentations

Proliferation of frameworks

□ Dynamo

□ Charlotte

□ Thunder

□ Sugar

TM

Concepts Overview

□ Real-time applications (Channels)

□ MVC

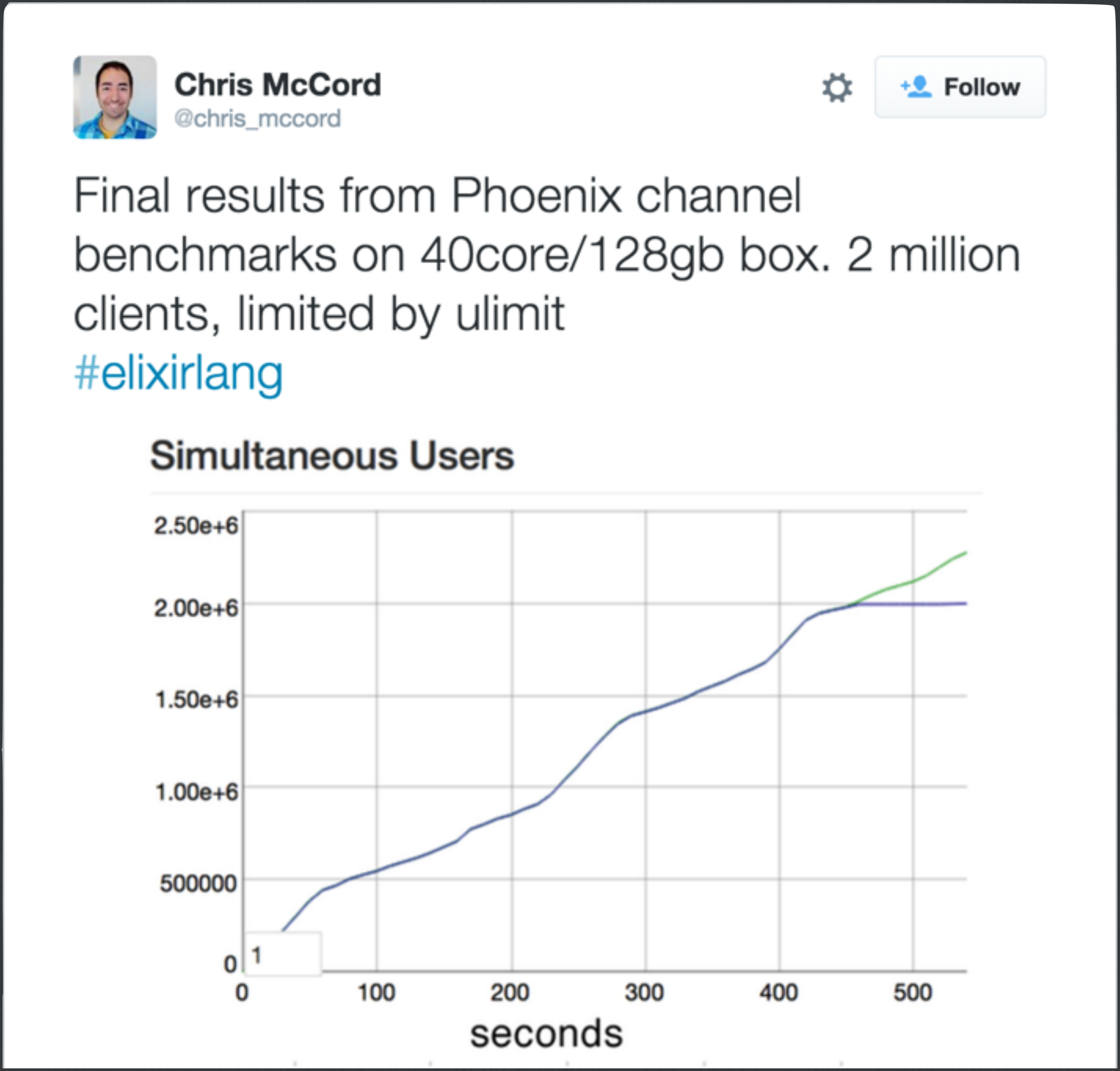
□ Middlewares (Plug)

Concepts Overview

- Pipelines
- Endpoints

Chat example

http://phoenixchat.herokuapp.com/



Chris McCord
@chris_mccord

Final results from Phoenix channel benchmarks on 40core/128gb box. 2 million clients, limited by ulimit
#elixirlang

Other great features

- “ORM” and Migrations through Ecto
- Generators
- Live reloading of assets
- Internationalization

“Phoenix is not Rails”

- □ Similarities
  - □ Productivity
  - □ MVC
  - □ Security
  - □ Test driven

- □ Differences
  - □ Applications
  - □ Explicit connection transformation
  - □ Channels
  - □ Assets (Brunch)

What’s for the future?

□ Presence channels

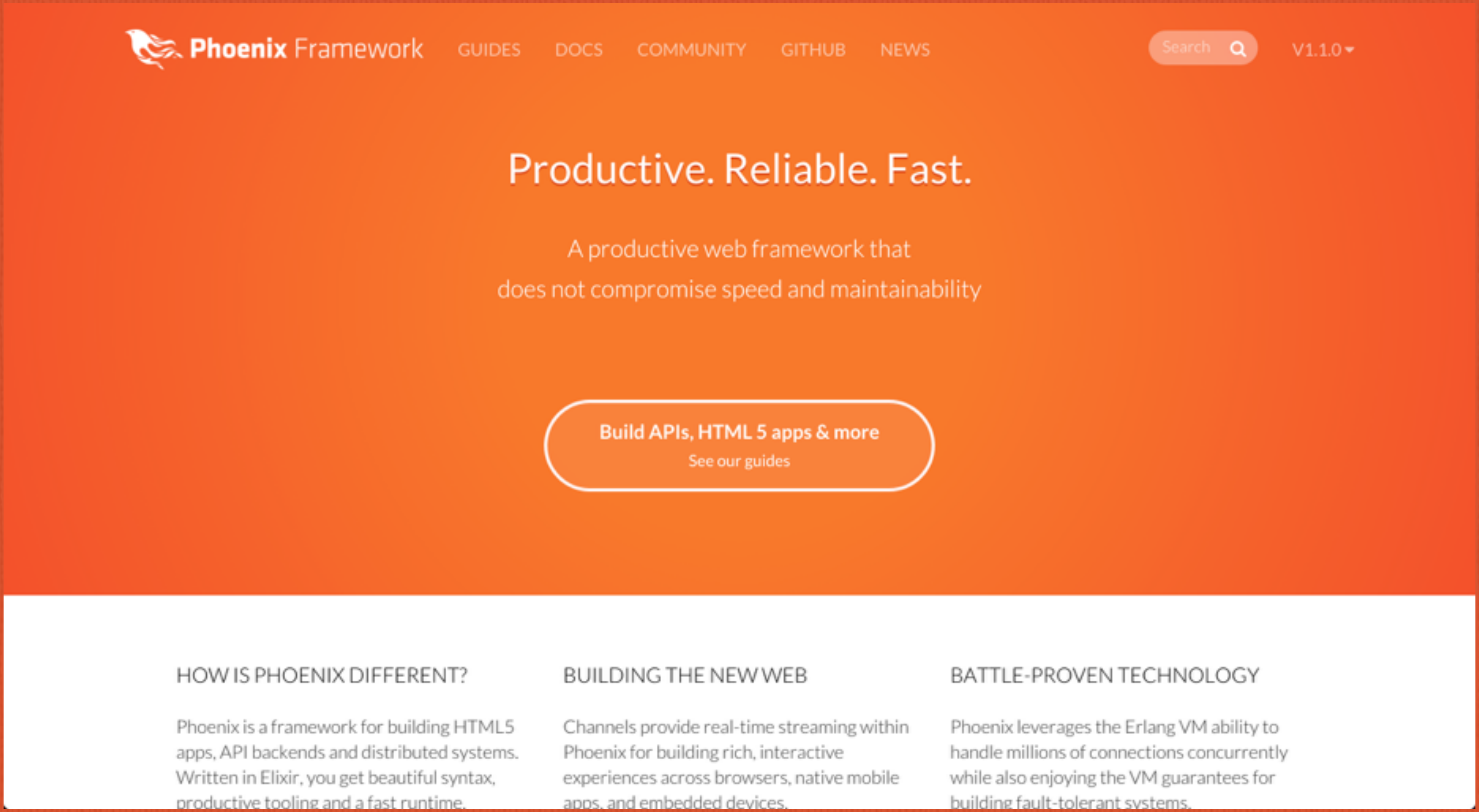
□ GraphQL

Not convinced?

- □ Fast and productive
- □ Easy to test
- □ Highly scalable
- □ “With Phoenix, nothing is global. There is no monolith”
- □ “Phoenix favors explicitness in most of its stack.”

Give it a shot!

- Modern
- Stable
- Great community



Phoenix Framework GUIDES DOCS COMMUNITY GITHUB NEWS Search V1.1.0

Productive. Reliable. Fast.

A productive web framework that
does not compromise speed and maintainability

Build APIs, HTML 5 apps & more
See our guides

HOW IS PHOENIX DIFFERENT?

Phoenix is a framework for building HTML5 apps, API backends and distributed systems. Written in Elixir, you get beautiful syntax, productive tooling and a fast runtime.

BUILDING THE NEW WEB

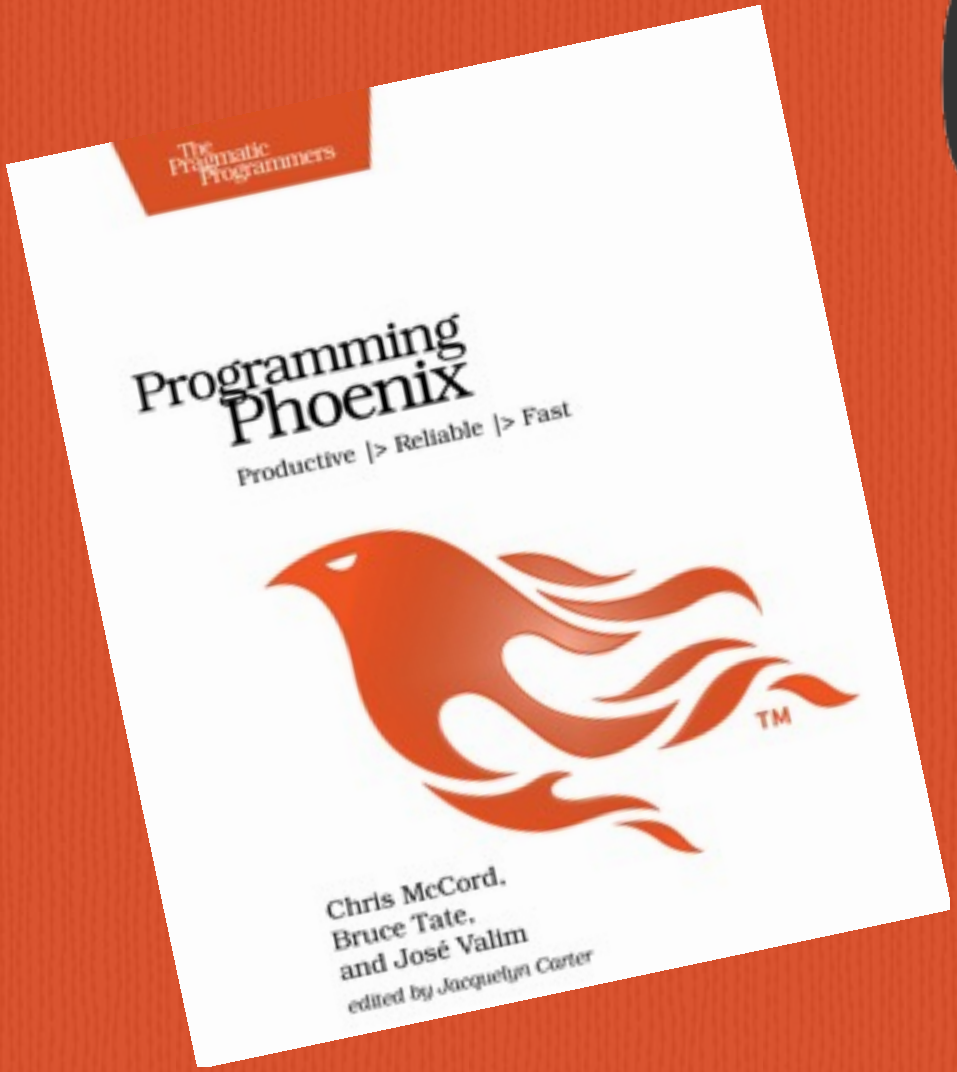
Channels provide real-time streaming within Phoenix for building rich, interactive experiences across browsers, native mobile apps, and embedded devices.

BATTLE-PROVEN TECHNOLOGY

Phoenix leverages the Erlang VM ability to handle millions of connections concurrently while also enjoying the VM guarantees for building fault-tolerant systems.

http://phoenixframework.org/

Confreaks


The Pragmatic Programmers

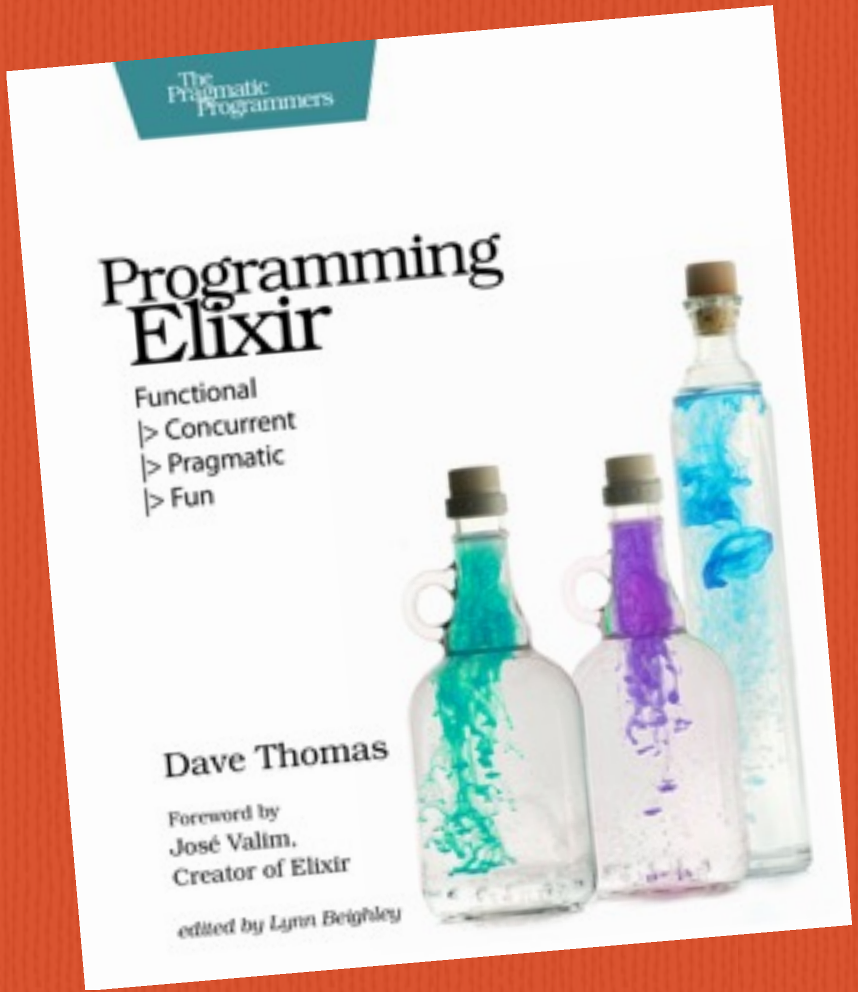
Programming Phoenix

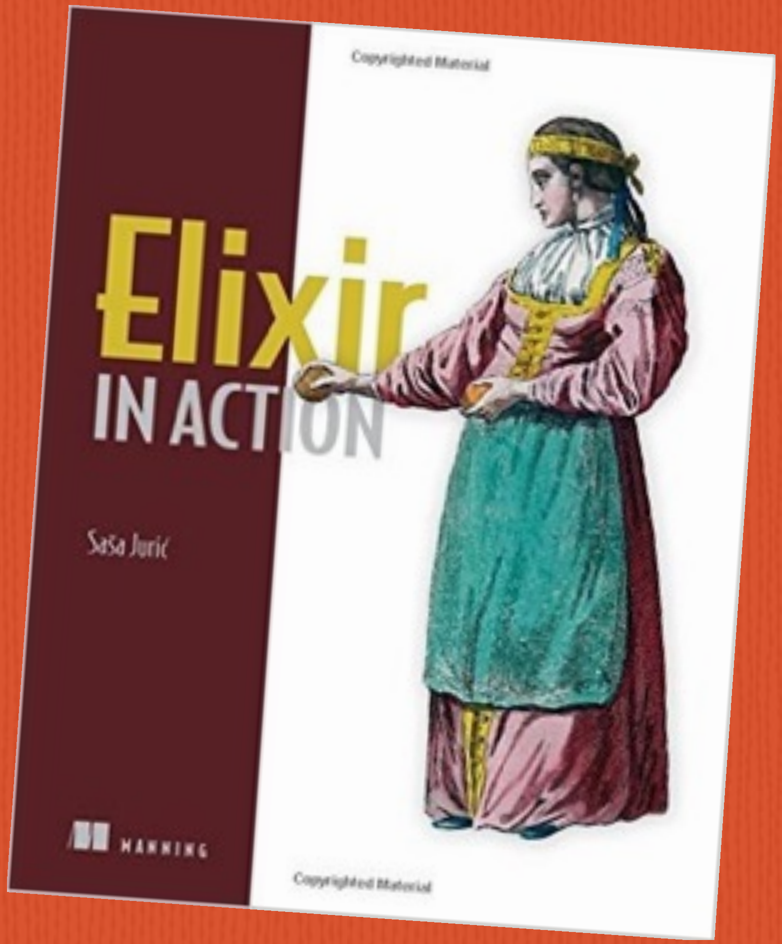
Productive |> Reliable |> Fast

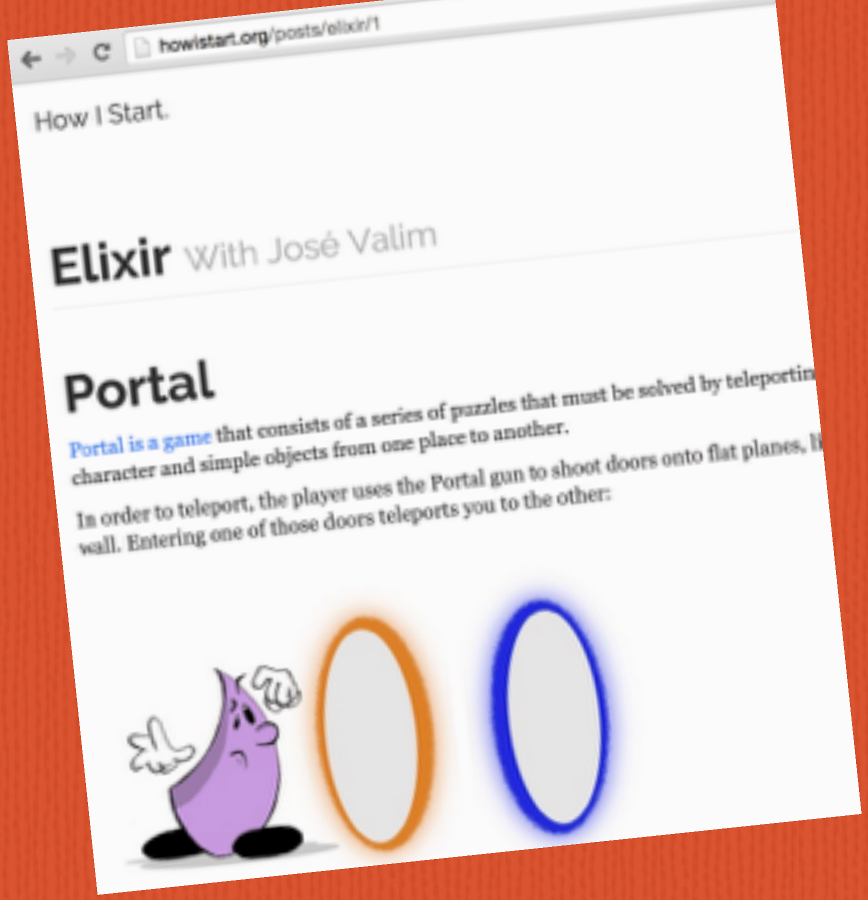
Chris McCord,
Bruce Tate,
and José Valim

edited by Jacquelyn Carter

DOCKYARD







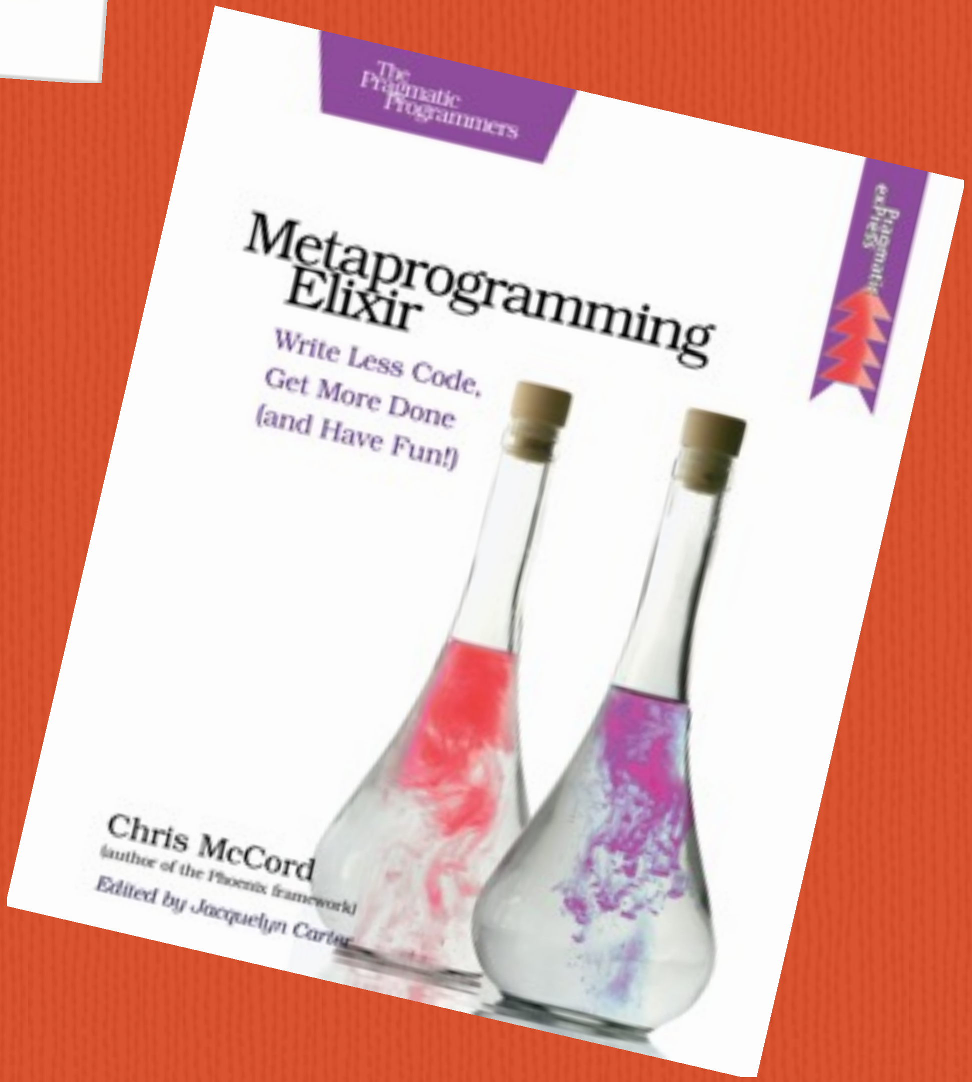
How I Start.

Elixir With José Valim

Portal

Portal is a game that consists of a series of puzzles that must be solved by teleportin character and simple objects from one place to another.

In order to teleport, the player uses the Portal gun to shoot doors onto flat planes, li wall. Entering one of those doors teleports you to the other:



Obrigado! Thanks! Köszi!

Questions?